MULTIMEDIA UNIVERSITY

STUDENT IDENTIFICATION NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2017/2018

BIC1024 – BUSINESS APPLICATIONS DEVELOPMENT

(All sections / Groups)

17 MARCH 2018
2.30 p.m – 4.30 p.m
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 7 pages with 6 Questions only.

2. Attempt **FOUR** out of FIVE questions in Part A. Attempt **ALL** questions in Part B. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

PART A: Answer FOUR questions only. (20 marks for each question)

QUESTION 1

i. Before developing and coding a computer program, it is important to have all programmers involved to agree upon the naming conventions of constants and variables to be used. Briefly discuss the reasons to follow a naming convention for constants and variables. (10 marks)

ii. Evaluate the following equations, given the values a = 1, b = 3, c = 5, d = True, and e = False.

a. z = a + b * b – c \ 2 (2 marks)
b. z = 5 * (a + b) MOD 2 (2 marks)
c. z = 3 * c / 5 + ((c + 1) / (a + 1)) (2 marks)
d. y = NOT (d AND e) OR d (2 marks)
e. y = NOT e AND c > b (2 marks)

(Total: 20 marks)

QUESTION 2

i. Your team has been assigned to develop a computer application that generate and print the total score of each participant after 4 attempts. The system will begin by asking each user to enter his/her name and score of each attempt. The system will calculate the total score and display the participant total score. This process will be repeated until all participants' scores have been recorded.

Your team has decided to develop the program using modular structured programming method. Before coding the program, develop an interactivity chart for this program. Provide proper labelling on the interactivity chart to ease communication among team members. (6 marks)

ii. Briefly explain automatic counter loop and the suitable situation to deploy this loop structure during computer program development. Draw a flowchart to represent automatic counter loop. (6 marks)

iii. Computer programmers may use call-by-value or call-by-reference parameter to send data from one module to another. Differentiate between the use of call-by-value and call-by-reference. (8 marks)

(Total: 20 marks)

Continued...

QUESTION 3

i. Your team has been assigned to develop a computer application that determines the performance of students during a test. After entering score of the test, the system will display a message based on the score. The following table shows the message that will be displayed based on the score obtained.

Score	Message
0 to 49	"There is a room for improvement."
50 to 74	"Keep it up!"
75 - 100	"Excellent!"

The team has decided to use modular structured programming method to develop the program. Each member will be given a module to create a portion of the program. You are assigned by your team leader to complete the portion of the program that decides the message to be displayed based on the score. You have decided to name the module as "Decide", and the positive decision logic will be used for this purpose.

a. Before coding the program, draw a flowchart to illustrate the message to be displayed based on the score obtained from another module. (8 marks)

b. By using C language, write the codes that represent the flowchart drawn in (a). Assume all variables has been declared and are set as global variables. (8 marks)

ii. Explain the use of recursion in modular structure programming. (4 marks)

(Total: 20 marks)

Continued...

QUESTION 4

i. The following portion of computer codes are written by using C language.

```
int n;

n = 20;

while(n < 70)
{
    printf(" %d \n", n);

    n = n + 9;
}
```

a. Illustrate the output of the codes above. (5 marks)

b. Rewrite the codes in (a) by using a "*for*" loop structure. Assume all variables have been declared. (5 marks)

ii. Describe how selection sort works. Given the unsorted numbers below, provide the steps and descriptions in performing selection sort to sort these numbers in ascending order. (10 marks)

23	43	33	13	75	53

(Total: 20 marks)

Continued...

QUESTION 5

The following arrays show the types of computer storages and its respective price.

product	Hard disk 250Gb	Hard disk 500Gb	Hard disk 1Tb	Hard disk 2Tb	Flash drive 16Gb	Flash drive 32Gb	Flash drive 64Gb
price	200.50	300.30	400.10	534.50	122.00	67.50	80.30

Attempt the following by using C language.

i. Declare 2 arrays with proper data type and name these as "product" and "price" to store the product name and price respectively. (5 marks)

ii. The owner has decided to reduce the price of "Hard disk 500Gb" by RM10.00. Write codes to update the price for this product. (2 marks)

iii. It seems there is an error on the name of "Flash drive 16Gb". It should be named as "Flash drive 128Gb". Create codes to replace or update the name of this product. (3 marks)

iv. The owner has decided to give 10 percent discount on all items in the list. Update the latest price for all products. Use a loop to complete this task. Assume all variables have been declared. (5 marks)

v. Using a loop structure, write codes to display all products' name and their respective prices. Assume all variables have been declared. (5 marks)

(Total: 20 marks)

Continued...

PART B: Answer ALL questions. (20 marks)

QUESTION 6

A grocery store owner is seeking your help to create a computer program to assist him in identifying products that are priced wrongly in the store. According to the owner, the company should be earning at least 30% of profit for every single product. However, a human error has caused certain product prices wrongly labelled. Hence, the owner needs your help to create a system that will calculate the percentage of profit for a particular product after he enters its cost and selling price. The system will then decide what actions to be taken. The following table shows the actions that will be carried out by the system based on the calculated percentage of profit.

Profit percentage (%)	Action(s)
More than or equal to 30%	"OK. Place product on sales."
More than or equal to 0%	- "Sales price adjustment needed" - Calculate the minimum selling price to achieve at least 30%) profit
Less than 0%	Nonprofit product!!!

A portion of the computer codes written in C language for this program is shown in the following pages. Attempt the following tasks:

i. What does the following function prototype (line 8) means? (2 marks)

```
void decision(float, float);
```

ii. Based on the decision module shown in the following page, draw a flowchart to represent this module. (6 marks)

iii. It seems that the NOT_OK() module is missing from the codes. When this module is called, the module will display a message stating "Sales price adjustment needed". This is followed by calculating the minimum selling price needed to achieve at least 30% profit. The formula to calculate the minimum price is as follows:

$$\text{Minimum selling price to achieve at least 30\% profit} = \frac{3*cost}{10} + cost$$

Write the codes for NOT_OK() module to complete this business application. (6 marks)

iv. Based on the given codes, draw an interactivity chart to represent this business application. (6 marks)

(Total: 20 marks)

Continued...

Programming Codes:

```
#include<stdio.h>

void input_selling(void);
void input_cost(void);
float calculate_percentage(void);
void display_OK(void);
float NOT_OK(void);
void decision(float, float);

float selling_price;
float cost;

main()
{

    float percent_profit;
    float adjusted_selling_price;

    input_selling();

    input_cost();

    percent_profit = calculate_percentage();

    decision(percent_profit, adjusted_selling_price);


    system("PAUSE");

    return 0;
}

void input_selling(void)
{
    printf("Enter cost of the product: ");
    scanf(" %f", &cost);

    return;
}
```

Continued...

Programming Codes: (Continued…)

```
40
41  void input_cost(void)
42  {
43      printf("Enter current selling price :");
44      scanf(" %f", &selling_price);
45
46      return;
47  }
48
49  float calculate_percentage(void)
50  {
51      float mpercent_profit;
52
53      mpercent_profit = (selling_price - cost) / cost * 100;
54
55      return mpercent_profit;
56  }
57
58  void decision(float mpercent_profit, float madjusted_selling_price)
59  {
60      if(mpercent_profit >= 30.0)
61      {
62          display_OK();
63      }
64      else if(mpercent_profit >= 0.0)
65      {
66          madjusted_selling_price = NOT_OK();
67
68          printf("Sales price should be at least RM %.2f \n", madjusted_selling_price);
69      }
70      else
71      {
72          printf("Non profit product!!!\n");
73      }
74
75      return;
76  }
77
78  void display_OK(void)
79  {
80      printf("OK. Place product on sales. \n");
81
82      return;
83  }
```

End of Paper